L 12

```
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
```

```
_$

Ps
--
YZ

Z$

Z$

Z$


Z$


Z$




Z$

Z$

Z$

Z$

Z$

Z$


Z$
```

```
**FILE**ID**SYSLKWSET                                      G 3



  SSSSSSSS  YY      YY    SSSSSSSS  LL          KK      KK  WW      WW    SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT
  SSSSSSSS  YY      YY    SSSSSSSS  LL          KK      KK  WW      WW    SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT
SS          YY      YY  SS          LL          KK      KK  WW      WW  SS          EE              TT
SS          YY      YY  SS          LL          KK      KK  WW      WW  SS          EE              TT
SS            YY  YY    SS          LL          KK    KK    WW      WW  SS          EE              TT
SS            YY  YY    SS          LL          KK    KK    WW      WW  SS          EE              TT
  SSSSSS        YY        SSSSSS    LL          KKKKKK      WW      WW    SSSSSS    EEEEEEEE        TT
  SSSSSS        YY        SSSSSS    LL          KKKKKK      WW      WW    SSSSSS    EEEEEEEE        TT
        SS      YY              SS  LL          KK    KK    WW  WW  WW          SS  EE              TT
        SS      YY              SS  LL          KK    KK    WW  WW  WW          SS  EE              TT
        SS      YY              SS  LL          KK      KK  WWWW  WWWW          SS  EE              TT          ....
        SS      YY              SS  LL          KK      KK  WWWW  WWWW          SS  EE              TT          ....
SSSSSSSS        YY      SSSSSSSS    LLLLLLLLLL  KK      KK  WW      WW  SSSSSSSS    EEEEEEEEEE      TT          ....
SSSSSSSS        YY      SSSSSSSS    LLLLLLLLLL  KK      KK  WW      WW  SSSSSSSS    EEEEEEEEEE      TT          ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000            .TITLE  SYSLKWSET - LOCK/UNLOCK FROM WORKING SET/MEMORY
0000            .IDENT  'V04-000'
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000
0000    ;++
0000    ; FACILITY:
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ; ENVIRONMENT:
0000    ;
0000    ; AUTHOR: PETER H. LIPMAN       , CREATION DATE: 24-APR-78
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-009 WMC0005         Wayne Cardoza           26-Feb-1984
0000    ;               Support for resident global sections.
0000    ;
0000    ;       V03-008 WMC0004         Wayne Cardoza           28-Nov-1983
0000    ;               Move kernel stack expansion WSL slots.
0000    ;
0000    ;       V03-007 WMC0003         Wayne Cardoza           13-Oct-1983
0000    ;               Add support for expanding kernel stack.
0000    ;
0000    ;       V03-006 LJK0247         Lawrence J. Kenah       1-Sep-1983
0000    ;               Remove fatal bugcheck when MMG$SCNWSLX fails. The global
0000    ;               page was removed from the working set and should simply be
0000    ;               faulted another time.
0000    ;
0000    ;       V03-005 TCM0001         Trudy C. Matthews       1-Apr-1983
0000    ;               Change references to working set fields in PHD so that
0000    ;               they are used as unsigned words.
0000    ;
0000    ;       V03-004 WMC0002         Wayne Cardoza           02-Mar-1983
0000    ;               MMG$CRECOM1 HAS GONE AWAY
```

```
0000     ;
0000     ;       V03-003 WMC0001         Wayne Cardoza           29-Sep-1982
0000     ;               Fix loop when trying to reduce working set to quota.
0000     ;
0000     ;       V03-002 KDM46395        Kathleen D. Morse       28-Jun-1982
0000     ;               Check whether page table pages should be moved out
0000     ;               of the locked portion of the working set when the
0000     ;               PTWSLELCK byte array count goes to zero.
0000     ;
0000     ;--
```

```
                 0000             .SBTTL  DECLARATIONS
                 0000      ;
                 0000      ; INCLUDE FILES:
                 0000      ;
                 0000             $IPLDEF                         ;PROCESSOR PRIORITY LEVELS
                 0000             $MMGDEF                         ; Offsets from FP into scratch area
                 0000             $OPDEF                          ; Define opcode equivalent values
                 0000             $PCBDEF                         ;PROCESS CONTROL BLOCK DEFINITIONS
                 0000             $PFNDEF                         ;PAGE FRAME NUMBER DATA BASE
                 0000             $PHDDEF                         ;PROCESS HEADER DEFINITIONS
                 0000             $PRDEF                          ;PROCESSOR REGISTER DEFINTIONS
                 0000             $PRVDEF                         ;PRIVILEGE BIT DEFINITIONS
                 0000             $PTEDEF                         ;PAGE TABLE ENTRY DEFINITIONS
                 0000             $SSDEF                          ;SYSTEM STATUS CODE DEFINTIONS
                 0000             $VADEF                          ;VIRTUAL ADDRESS VIELDS
                 0000             $WQHDEF                         ;WAIT QUEUE HEADER
                 0000             $WSLDEF                         ;WORKING SET LIST ENTRY DEFINTIONS
                 0000      ;
                 0000      ; EXTERNAL SYMBOLS:
                 0000      ;
                 0000      ;
                 0000      ; MACROS:
                 0000      ;
                 0000      ;
                 0000      ; EQUATED SYMBOLS:
                 0000      ;
                 0000      ; OFFSET FROM AP
                 0000      ;
        00000004 0000             INADR           = 4             ;OFFSET TO INPUT RANGE
        00000008 0000             RETADR          = 8             ;OFFSET TO RETURN RANGE
        0000000C 0000             ACMODE          = 12            ;ACCESS MODE
                 0000      ;
                 0000      ; BIT DEFINITIONS IN R8 CONTROL LONG WORD
                 0000      ;
                 0000             _VIELD  LCK,8,<-
                 0000                     <LCKPAG,,M>,-           ;SET IF LOCKING, CLEAR IF UNLOCKING
                 0000                     <GLOBAL,,M> -           ;SET IF GLOBAL PAGE
                 0000                     >
                 0000      ;
                 0000      ; OWN STORAGE:
                 0000      ;
             00000000             .PSECT  $$$210,LONG
                 0000
                 0000      MMG$GL_PFNLOCK::
        40000000 0000             .LONG   1@30                    ;DOWN COUNTER OF PAGES REMAINING
                 0004                                             ;THAT MAY BE LOCKED IN MEMORY
```

```
0004          .SBTTL  LOCK/UNLOCK PAGES IN WORKING SET/MEMORY
0004    ;++
0004    ; FUNCTIONAL DESCRIPTION:
0004    ;
0004    ;       THE LKWSET SYSTEM SERVICE LOCKS THE SPECIFIED RANGE OF PAGES
0004    ; INTO THE WORKING SET.  THE PAGES LOCKED ARE NO LONGER CANDIDATES
0004    ; FOR PAGE REPLACEMENT AND ARE THUS GUARANTEED RESIDENT IN MEMORY WHEN
0004    ; THE PROCESS IS IN THE BALANCE SET.
0004    ;       SINCE SOME REASONABLE NUMBER OF DYNAMIC PAGES IS REQUIRED IN ORDER
0004    ; TO GUARANTEE THAT AN INSTRUCTION CAN RUN TO COMPLETION, THE NUMBER OF
0004    ; PAGES THAT MAY BE LOCKED IN THE WORKING SET IS LIMITED.
0004    ;       THE ULWSET SYSTEM SERVICE UNLOCK THE PAGES FROM THE WORKING SET.
0004    ;
0004    ; CALLING SEQUENCE:
0004    ;
0004    ;       CALLG   ARGLIST,G^SYS$LKWSET
0004    ;       CALLG   ARGLIST,G^SYS$ULWSET
0004    ;       CALLG   ARGLIST,G^SYS$LCKPAG
0004    ;       CALLG   ARGLIST,G^SYS$ULKPAG
0004    ;
0004    ; INPUT PARAMETERS:
0004    ;
0004    ;       INADR(AP) = ADDRESS OF 2 LONG WORDS THE 1ST OF WHICH SPECIFIES
0004    ;               THE STARTING VIRTUAL ADDRESS TO LOCK/UNLOCK, THE 2ND SPECIFIES
0004    ;               THE ENDING VIRTUAL ADDRESS TO LOCK/UNLOCK (INCLUSIVE).
0004    ;       RETADR(AP) = ADDRESS OF A 2 LONGWORD ARRAY INTO WHICH IS RETURNED
0004    ;               THE STARTING AND ENDING VIRTUAL ADDRESSES (INCLUSIVE)
0004    ;               OF THE PAGES JUST LOCKED/UNLOCKED
0004    ;       ACMODE(AP) = THE ACCESS MODE (MAXIMIZED WITH CALLING MODE)
0004    ;               ON BEHALF OF WHICH THE LOCK/UNLOCK IS TO BE DONE
0004    ;               THIS MODE MUST BE AT LEAST AS PRIVILEGED AS THE MODE
0004    ;               OF THE PAGE OWNER.
0004    ;
0004    ; IMPLICIT INPUTS:
0004    ;
0004    ;       NONE
0004    ;
0004    ; OUTPUT PARAMETERS:
0004    ;
0004    ;       R0 = SYSTEM STATUS CODE
0004    ;
0004    ; IMPLICIT OUTPUTS:
0004    ;
0004    ;       NONE
0004    ;
0004    ; COMPLETION CODES:
0004    ;
0004    ;       SS$_WASSET                              ;PREVIOUS STATE OF LOCK WAS SET
0004    ;       SS$_WASCLR                              ;PREVIOUS STATE OF LOCK WAS CLEAR
0004    ;       SS$_ACCVIO                              ;ACCESS VIOLATION
0004    ;       SS$_NOPRIV                              ;NO PRIVILEGE TO LOCK/UNLOCK PAGE
0004    ;       SS$_LKWSETFUL                           ;LOCKED PORTION OF WORKING SET IS FULL
0004    ;       SS$_LCKPAGFUL                           ;NO MORE PAGES MAY BE LOCKED IN MEMORY
0004    ;       SS$_PAGOWNVIO                           ;PAGE OWNER VIOLATION
0004    ;
0004    ; SIDE EFFECTS:
0004    ;
```

```
                        0004  ;       NONE
                        0004  ;
                        0004  ;--
                        0004
                        0004  ;
                        0004  ;       *********************************************************************
                        0004  ;
                        0004  ;       **************** THE FOLLOWING CODE MAY BE PAGED ********************
                        0004  ;
                    00000000          .PSECT  Y$EXEPAGED
                        0000  ;
                        0000  ;       *********************************************************************
                        0000  ;
                        0000          .ENABL  LSB
                        0000
                  01FC  0000          .ENTRY  EXE$ULWSET,^M<R2,R3,R4,R5,R6,R7,R8>
                        0002
          58   20   9A  0002          MOVZBL  #WSL$M_WSLOCK,R8        ;UNLOCKING FROM WORKING SET
               05   11  0005          BRB     10$
                        0007
                  01FC  0007          .ENTRY  EXE$ULKPAG,^M<R2,R3,R4,R5,R6,R7,R8>
                        0009
          58   10   9A  0009          MOVZBL  #WSL$M_PFNLOCK,R8       ;UNLOCKING FROM MEMORY
          50   09   3C  000C  10$:    MOVZWL  #SS$_WASSET,R0          ;INIT TO 'PAGES WERE LOCKED'
          5E   1C   C2  000F          SUBL    #-MMG$C_LENGTH,SP       ;RESERVE SCRATCH AREA
                        0012          SAVIPL                          ;SAVE CALLERS IPL FOR EXIT
               27   11  0015          BRB     30$
                        0017
                  01FC  0017          .ENTRY  EXE$LKWSET,^M<R2,R3,R4,R5,R6,R7,R8>
                        0019
          58   20   9A  0019          MOVZBL  #WSL$M_WSLOCK,R8        ;LOCKING IN WORKING SET
               0E   11  001C          BRB     20$
                        001E
                  01FC  001E          .ENTRY  EXE$LCKPAG,^M<R2,R3,R4,R5,R6,R7,R8>
                        0020
          50   24   3C  0020          MOVZWL  #SS$_NOPRIV,R0          ;ASSUME CANNOT LOCK PAGES IN MEMORY
                        0023          IFNPRIV PSWAPM,45$              ;BRANCH IF CANNOT LOCK PAGES IN MEMORY?
          58   10   9A  0029          MOVZBL  #WSL$M_PFNLOCK,R8       ;LOCKING IN MEMORY
          5E   1C   C2  002C  20$:    SUBL    #-MMG$C_LENGTH,SP       ;RESERVE SCRATCH AREA
                        002F          DSBINT  #IPL$_ASTDEL            ;PREVENT SYSTEM WORKING SET GROWTH AST
               41   10  0035          BSBB    50$                     ;SET QUOTA WORKING SET SIZE BEFORE LOCK
          50   01   3C  0037  25$:    MOVZWL  #SS$_WASCLR,R0          ;INIT TO 'PAGES WERE UNLOCKED'
       00 58   08   E2  003A          BBSS    #LCK_V_LCKPAG,R8,30$    ;SET LOCK FLAG
 56   00000011'EF   DE  003E  30$:    MOVAL   MMG$LCKULKPAG,R6        ;ADDRESS OF PAGE LOCK/UNLOCK ROUTINE
58   10   10   50   F0  0045          INSV    R0,#16,#16,R8           ;INITIAL STATUS IN HIGH 16 BITS
            FFB3'   30  004A          BSBW    MMG$INADRINI            ;INITIALIZE RETURN ADDRESS ARAYS & SCRATCH A
            27 50   E9  004D          BLBC    R0,45$
          52   54   7D  0050          MOVQ    R4,R2                   ;R2 = START OF RANGE, R3 = END
            FFAA'   30  0053          BSBW    MMG$CREDEL              ;COMMON CREATE CODE
               50   DD  0056          PUSHL   R0                      ;SAVE STATUS
            FFA5'   30  0058          BSBW    MMG$RETRANGE            ;RETURN AFFECTED ADDRESS RANGE
            19 50   E9  005B          BLBC    R0,45$                  ;USE THIS BAD STATUS RATHER THAN CREDEL
               50   BA  005E          POPR    R0
            11 50   E8  0060          BLBS    R0,40$                  ;BRANCH IF SUCCESSFUL
     50   0194 8F   B1  0063          CMPW    #SS$_LKWSETFUL,R0       ;IS THIS LOCKED WORKING SET FULL STATUS
               0A   12  0068          BNEQ    40$                     ;BRANCH IF NOT
               0C   10  006A          BSBB    50$                     ;REVERIFY SYSTEM DIDN'T SHRINK LIMIT
```

```
                C8 52   E8  006C   230          BLBS    R2,25$                          ;BRANCH IF ADJUSTMENT OCCURED
        50   0194 8F   3C  006F   231          MOVZWL  #SS$_LKWSETFUL,R0               ;RESET ERROR RETURN STATUS
                            0074   232 40$:     ENBINT                                  ;RESTORE CALLERS IPL
                        04  0077   233 45$:     RET
                            0078   234
   55     00000000'GF   D0  0078   235 50$:     MOVL    G^CTL$GL_PHD,R5                 ;GET PROCESS HEADER ADDRESS
                   52   D4  007F   236          CLRL    R2                              ;ASSUME NO ADJUSTMENT RETURN STATUS
 50    18 A5     08 A5  A3  0081   237 55$:     SUBW3   PHD$W_WSLIST(R5),PHD$W_WSQUOTA(R5),R0 ;GET MAX LOCKABLE SIZE
                   50   B6  0087   238          INCW    R0                              ;IN ORIGIN 1
             50    50   3C  0089   239          MOVZWL  R0,R0                           ;GET IT IN A LONGWORD
        7E     50 A5   3C  008C   240          MOVZWL  PHD$W_WSSIZE(R5),-(SP)          ;GET CURRENT SIZE
             50    8E   C2  0090   241          SUBL    (SP)+,R0                        ;GET DIFFERENCE FROM CURRENT SIZE
                   1B   13  0093   242          BEQL    60$                             ;BRANCH IF CURRENT SIZE IS QUOTA, R0=0
        7E     50 A5   B0  0095   243          MOVW    PHD$W_WSSIZE(R5),-(SP)          ;SAVE STARTING SIZE
                   7E   D4  0099   244          CLRL    -(SP)                           ;NO RETURN SIZE NEEDED
             7E    50   32  009B   245          CVTWL   R0,-(SP)                        ;FORM ADJUSTMENT SIZE
    00000000'EF    02   FB  009E   246          CALLS   #2,SYS$ADJWSL                   ;ADJUST THE WORKING SET SIZE
        8E     50 A5   B1  00A5   247          CMPW    PHD$W_WSSIZE(R5),(SP)+          ;DID SIZE CHANGE
                   05   13  00A9   248          BEQL    60$                             ;NO - GIVE UP
             52    01   88  00AB   249          BISB    #1,R2                           ;SET ADJUSTMENT PERFORMED
                   D1   11  00AE   250          BRB     55$                             ;DO THIS TILL SIZE AT AUTHORIZED QUOTA
                        05  00B0   251 60$:     RSB                                     ;R0 LOW BIT INDICATES CHANGE OCCURRED
                            00B1   252
                            00B1   253          .DSABL  LSB
```

```
                              00B1         .SBTTL  UNLOCK GLOBAL PAGE FROM WORKING SET
                              00B1  ;
                              00B1  ; CALLING SEQUENCE:
                              00B1  ;
                              00B1  ;      BSBW    MMG$ULKGBLWSLE
                              00B1  ;
                              00B1  ; INPUTS:
                              00B1  ;
                              00B1  ;      IPL = SYNCH
                              00B1  ;      R1 = WORKING SET LIST INDEX OF ENTRY TO UNLOCK (HIGH 16 BITS = 0)
                              00B1  ;      R2 = WORKING SET LIST ENTRY (VA AND PAGE TYPE)
                              00B1  ;      R3 = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY
                              00B1  ;      R4 = PROCESS CONTROL BLOCK ADDRESS
                              00B1  ;      R5 = PROCESS HEADER ADDRESS (P1 SPACE OK)
                              00B1  ;
                              00B1  ; OUTPUTS:
                              00B1  ;
                              00B1  ;      R2 PRESERVED
                              00B1  ;      R8 IS ALTERED
                              00B1  ;      PHD$W_WSDYN(R5) = NEW WORKING SET LIST INDEX FOR THE UNLOCKED PAGE
                              00B1  ;
                              00B1  ; *********************************************************************
                              00B1  ;
                              00B1  ; *************** THE FOLLOWING CODE MUST BE RESIDENT ****************
                              00B1  ;
                          00000000         .PSECT  $MMGCOD
                              0000  ;
                              0000  ; *********************************************************************
                              0000  ;
                              0000 MMG$ULKGBLWSLE::
                   08    DD   0000         PUSHL   #IPL$_SYNCH              ;SAVE IPL FOR ENBINT TO RESTORE
             7E    51    7D   0002         MOVQ    R1,-(SP)                 ;PUSHL VA, PUSHL WSLX
             50    63    D0   0005         MOVL    (R3),R0                  ;GET PAGE TABLE ENTRY
 58   00090030 8F    D0   0008         MOVL    #<SS$_WASSET @16 ! -     ;SET UNLOCK PARAMETER
                              000F                 WSL$M_WSLOCK ! WSL$M_PFNLOCK>,R8 ;IN R8
                   43    11   000F         BRB     ULKGBLWSLE               ;JOIN COMMON LOCK/UNLOCK CODE
```

```
0011            .SBTTL  LOCK/UNLOCK SINGLE PAGE IN WORKING SET/MEMORY
0011    ;++
0011    ; FUNCTIONAL DESCRIPTION:
0011    ;
0011    ;
0011    ; CALLING SEQUENCE:
0011    ;
0011    ;       BSBW    MMG$LCKULKPAG
0011    ;
0011    ; INPUT PARAMETERS:
0011    ;
0011    ;       R0          = ACCESS MODE TO CHECK AGAINST PAGE OWNER
0011    ;       R2          = VIRTUAL ADDRESS
0011    ;       R4          = PCB ADDRESS
0011    ;       R5          = PROCESS HEADER ADDRESS - P1 OR SYSTEM SPACE
0011    ;       R6          = COUNT - 1 OF PAGES REMAINING TO  BE DELETED INCLUDING THIS ONE
0011    ;       R7          = +^X200 IF GOING FORWARD IN ADDRESS SPACE
0011    ;                   = -^X200 IF GOING BACKWARDS IN ADDRESS SPACE
0011    ;       R8<0:7>     = WSL$M_WSLOCK IF LOCKING IN WORKING SET
0011    ;                   = WSL$M_PFNLOCK IF LOCKING IN PHYSICAL MEMORY
0011    ;                   = EITHER OR BOTH OF THE ABOVE IF UNLOCKING
0011    ;       R8<8>       = LCK_V_LCKPAG SET IF LOCKING, CLEAR IF UNLOCKING
0011    ;       R8<16:31> = SS$_WASSET OR SS$_WASCLR
0011    ;
0011    ; IMPLICIT INPUTS:
0011    ;
0011    ;       NONE
0011    ;
0011    ; OUTPUT PARAMETERS:
0011    ;
0011    ;       R0          = SYSTEM STATUS CODE (INCLUDING 'WAS SET' BIT IF SUCCESSFUL)
0011    ;       R2          - PRESERVED
0011    ;       R8<16:31> = SS$_WASSET IF LOCKING AND THE LOCK HAD ALREADY BEEN SET
0011    ;                   = SS$_WASCLR IF UNLOCKING AND THE LOCK HAD ALREADY BEEN CLEAR
0011    ;                   - OTHERWISE IT IS PRESERVED.
0011    ;
0011    ; IMPLICIT OUTPUTS:
0011    ;
0011    ;       NONE
0011    ;
0011    ; COMPLETION CODES:
0011    ;
0011    ;       SS$_WASSET                          ;IF LOCKING, WAS ALREADY LOCKED
0011    ;       SS$_WASCLR                          ;IF UNLOCKING, WAS ALREADY UNLOCKED
0011    ;       SS$_ACCVIO                          ;ACCESS VIOLATION
0011    ;       SS$_PAGOWNVIO                       ;PAGE OWNER VIOLATION
0011    ;       SS$_LKWSETFUL                       ;IF LOCKING, LOCKED PORTION OF WORK SET FULL
0011    ;
0011    ; SIDE EFFECTS:
0011    ;
0011    ;       NONE
0011    ;
0011    ;--
```

```
                               0011  ;
                               0011  ; ********************************************************************
                               0011  ;
                               0011  ; **************** THE FOLLOWING CODE MUST BE RESIDENT *****************
                               0011  ;
                           00000011          .PSECT  $MMGCOD
                               0011  ;
                               0011  ; ********************************************************************
                               0011  ;
                               0011          .ENABL  LSB
                               0011
                               0011  MMG$LCKULKPAG::
                  51       D4  0011          CLRL    R1                      ;NO WORKING SET LIST INDEX INPUT
            00 58    09    E4  0013          BBSC    #LCK_V_GLOBAL,R8,15$    ;ASSUME NOT A GLOBAL PAGE
        62     01    00    0C  0017  15$:    PROBER  #0,#T,(R2)              ;IS THIS PAGE ACCESSIBLE?
                     04    12  001B          BNEQ    30$                     ;BRANCH IF YES
              50     0C    3C  001D          MOVZWL  #SS$_ACCVIO,R0          ;ACCESS VIOLATION IF NOT
                           05  0020          RSB
                  62       95  0021  30$:    TSTB    (R2)                    ;FAULT THE PAGE INTO THE WORKING SET
                               0023  ;
                               0023  ; R0 = ACCESS MODE FOR PAGE OWNER CHECK
                               0023  ; R1 = 0 INDICATING NO INPUT WORKING SET LIST INDEX
                               0023  ; R2 = VIRTUAL ADDRESS
                               0023  ;
                               0023          DSBINT  #IPL$_SYNCH             ;RAISE TO SYNCH TO INTERLOCK PFN DATA
                               0029                                          ;PUSHING CALLERS IPL
                     07    BB  0029          PUSHR   #^M<R0,R1,R2>           ;PUSH VA, LOCK/UNLOCK FLAG, ACCESS MODE
                  FFD2'    30  002B          BSBW    MMG$SVAPTECHK           ;RETURN R3 = SYS VA OF PAGE TABLE ENTRY
              50     63    D0  002E          MOVL    (R3),R0                 ;R0 = PAGE TABLE ENTRY
                     1A    19  0031          BLSS    50$                     ;BRANCH IF IT'S VALID
                               0033  ;
                               0033  ; PAGE NOT VALID!!  MUST HAVE SWAPPED JUST AFTER TOUCHING IT, TRY AGAIN
                               0033  ;
                     07    BA  0033  35$:    POPR    #^M<R0,R1,R2>
                               0035          ENBINT                          ;SO WE CAN FAULT IT AGAIN
                     E7    11  0038          BRB     30$                     ;TOUCH THE PAGE AGAIN
                               003A
        50     01EC 8F     3C  003A  40$:    MOVZWL  #SS$_PAGOWNVIO,R0       ;PAGE OWNER VIOLATION
                     03    11  003F          BRB     44$
              50     24    3C  0041  42$:    MOVZWL  #SS$_NOPRIV,R0          ;NO PRIVILEGE
                   0144    31  0044  44$:    BRW     110$
                               0047  ;
                               0047  ; THIS PAGE IS NOW VALID AND IN THE WORKING SET
                               0047  ;
                               0047
              50     09    3C  0047  47$:    MOVZWL  #SS$_WASSET,R0          ;SUCCESSFUL RETURN PATH FOR SHARED
                   013E    31  004A          BRW     110$                    ; MEMORY AND PFN MAPPED PAGES
                               004D
   8E     50     02   17   ED  004D  50$:    CMPZV   #PTE$V_OWN,#PTE$S_OWN,R0,(SP)+ ;CHECK PAGE OWNERSHIP
                     E6    19  0052          BLSS    40$                     ;BRANCH IF OWNER VIOLATION
                               0054  ULKGBLWSLE:
           EF 50     15    E0  0054          BBS     #PTE$V_WINDOW,R0,47$    ;RETURN SUCCESS FOR PFN MAPPED PAGES
   50     50     15   00   EF  0058          EXTZV   #PTE$V_PFN,#PTE$S_PFN,R0,R0 ;R0 = PAGE FRAME NUMBER
     00000000'EF     50    D1  005D          CMPL    R0,MMG$GL_MAXPFN        ;CHECK FOR PAGE IN SHARED MEMORY
                     E1    1A  0064          BGTRU   47$                     ;RETURN SUCCESS FOR SUCH PAGES
          53     00C8 C5   C2  0066          SUBL    PHD$L_P0BR(R5),R3       ;BYTE OFFSET OF PAGE TABLE ENTRY
    52    53     F7  8F    78  006B          ASHL    #-9,R3,R2               ;BYTE INDEX OF CONTAINING PAGE TABLE
```

```
                                  0070   403
                                  0070   404          ASSUME  PFN$C_PROCESS EQ 0
                                  0070   405          ASSUME  PFN$C_SYSTEM  EQ 1
                                  0070   406          ASSUME  PFN$C_GLOBAL  EQ 2
                                  0070   407          ASSUME  PFN$C_GBLWRT  EQ 3
                                  0070   408          ASSUME  PFN$C_PPGTBL  EQ 4
                                  0070   409          ASSUME  PFN$C_GPGTBL  EQ 5
                                  0070   410
               03   00      EE    0070   411          EXTV    #PFN$V_PAGTYP,#PFN$S_PAGTYP,- ;SEE IF PAGE IS A TYPE
        51   0000'DF40            0073   412                  @W^PFN$AB_TYPE[R0],R1    ;THAT CAN BE LOCKED
                    5E      13    0078   413          BEQL    56$                      ;PROCESS PAGES MAY BE LOCKED
           00 58    09      E2    007A   414          BBSS    #LCK_V_GLOBAL,R8,52$     ;PAGE IS GLOBAL (OR ERROR)
              02    51      D1    007E   415 52$:     CMPL    R1,#PFN$C_GLOBAL         ;SEE IF PAGE IS GLOBAL OR GLOBAL WRITE
                    0A      13    0081   416          BEQL    54$                      ;BRANCH IF GLOBAL READ ONLY
                    BC      19    0083   417          BLSS    42$                      ;BRANCH IF PAGE TYPE NOT LOCKABLE
                                  0085   418 ;
                                  0085   419 ; PAGE IS GLOBAL WRITE, MAY BE LOCKED IN MEMORY, BUT NOT IN WORKING SET BECAUSE
                                  0085   420 ; THE SWAPPER DROPS GLOBAL WRITABLE PAGES FROM THE WORKING SET ON OUTSWAP
                                  0085   421 ;
           04 58    08      E1    0085   422          BBC     #LCK_V_LCKPAG,R8,54$     ;OK IF UNLOCKING
           B4 58    05      E0    0089   423          BBS     #WSL$V_WSLOCK,R8,42$     ;CANNOT LOCK GLOBAL WRITE IN WORKING SET
              51    6E      3C    008D   424 54$:     MOVZWL  (SP),R1                  ;NORMALLY 0, WSLX IF ULKGBLWSLE
                    4C      12    0090   425          BNEQ    57$                      ;BRANCH IF IT IS A WSLX
                                  0092   426 ;
                                  0092   427 ; NEED TO SCAN FOR THE WORKING SET LIST INDEX SINCE THE WSLX ARRAY
                                  0092   428 ; DOES NOT CONTAIN THE BACKPOINTER FOR GLOBAL PAGES
                                  0092   429 ;
                    52      DD    0092   430          PUSHL   R2                       ;SAVE INDEX OF PAGE TABLE CONTAINING THIS PA
        52     08 AE        D0    0094   431          MOVL    8(SP),R2                 ;RECOVER VIRTUAL ADDRESS
                  0157      30    0098   432          BSBW    MMG$SCNWSLX              ;SCAN FOR WORKING SET LIST INDEX
                                  009B   433 ;
                                  009B   434 ; MAY HAVE BEEN SWAPPED, RECALCULATE PTE ADR, REFETCH PFN
                                  009B   435 ;
               15   00      EF    009B   436          EXTZV   #PTE$V_PFN,#PTE$S_PFN,-  ;REFETCH PFN GIVEN THE
        50   00C8 D543            009E   437                  @PHD$L_P0BR(R5)[R3],R0   ;BYTE OFFSET OF PTE IN PAGE TABLE
                    04      BA    00A3   438          POPR    #^M<R2>                  ;RECOVER SAVE R2
                    51      D5    00A5   439          TSTL    R1                       ;FOUND THE WORKING SET LIST INDEX?
                    35      12    00A7   440          BNEQ    57$                      ;BRANCH IF YES
                                  00A9   441 ;
                                  00A9   442 ; CHECK FOR A RESIDENT GLOBAL SECTION PAGE
                                  00A9   443 ;
        51   0000'DF40      D0    00A9   444          MOVL    @W^PFN$AL_BAK[R0],R1     ;GET BACKING STORE
           1F 51    16      E1    00AF   445          BBC     #PTE$V_TYP0,R1,55$       ;MUST BE PAGE-FILE
              51    51      32    00B3   446          CVTWL   R1,R1                    ;ISOLATE GLOBAL SECTION TABLE INDEX
     50   00000000'FF       DE    00B6   447          MOVAL   @L^MMG$GL_SYSPHD,R0      ;ADDRESS OF SYSTEM HEADER
           50    20 A0      C0    00BD   448          ADDL    PHD$L_PSTBASOFF(R0),R0   ;GET SECTION TABLE BASE
           50    6041       DE    00C1   449          MOVAL   (R0)[R1],R0              ;GET THE CORRECT SECTION TABLE ENTRY
03 0000'C0   00000000'8F    E1    00C5   450          BBC     #SEC$V_RESIDENT,SEC$W_FLAGS(R0),55$
                  FF75      31    00CF   451          BRW     47$                      ;RETURN SUCCESS
                                  00D2   452 ;
                                  00D2   453 ; THE GLOBAL PAGE WAS REMOVED FROM THE WORKING SET. THE ROUTINE WILL BE
                                  00D2   454 ; BACKED OUT AFTER THE STACK IS PUT INTO A CONSISTENT STATE.
                                  00D2   455 ;
                 FC AD      DD    00D2   456 55$:     PUSHL   MMG$L_MAXACMODE(FP)      ;PUSH ACCESS MODE FOR OWNER CHECK
                  FF5B      31    00D5   457          BRW     35$                      ;JOIN COMMON RETRY LOGIC
                                  00D8   458 ;
                                  00D8   459 ; PROCESS PAGE, GET WORKING SET LIST INDEX FROM PFN DATA BASE
```

```
                             00D8  ;
                             00D8  56$:              PFN_REFERENCE   -
                             00D8           MOVZWL   <@W^PFN$Ax_WSLX[R0],R1>,-        ;R1 = INDEX TO WORKING SET LIST ENTR
                             00D8                    LONG_OPCODE=MOVL,-
                             00D8                    IMAGE=SYS_NONPAGED
          53   6541     DE   00DE  57$:     MOVAL    (R5)[R1],R3                      ;R3 = ADDRESS OF ENTRY
                             00E2  ;
                             00E2  ; R0 = PAGE FRAME NUMBER
                             00E2  ; R1 = WORKING SET LIST INDEX FOR THIS PAGE
                             00E2  ; R2 = INDEX OF PAGE TABLE CONTAINING THIS PAGE
                             00E2  ; R3 = WORKING SET LIST ENTRY ADDRESS
                             00E2  ; 0(SP) = SCRATCH
                             00E2  ; 4(SP) = SAVED VIRTUAL ADDRESS
                             00E2  ;
       58 58     08     E1   00E2  58$:     BBC      #LCK_V_LCKPAG,R8,80$     ;BRANCH IF UNLOCKING
                             00E6  ;
                             00E6  ; LOCK THE PAGE IN THE WORKING SET LIST
                             00E6  ;
            63   58     93   00E6           BITB     R8,(R3)                          ;ALREADY LOCKED?
                 07     13   00E9           BEQL     60$                              ;BRANCH IF NOT
   58   10   10  09     F0   00EB           INSV     #SS$_WASSET,#16,#16,R8           ;PREVIOUS STATE OF LOCK WAS SET
                 78     11   00F0           BRB      96$                              ;NOTE INSV CLEARS Z BIT
            CF   8F     8B   00F2  60$:     BICB3    #^C<WSL$M_WSLOCK ! WSL$M_PFNLOCK>,- ;SAVE THE CURRENT STATE
          6E     63          00F5                    (R3),(SP)                        ;OF THE LOCK BITS
                 0E     12   00F7           BNEQ     62$                              ;BRANCH IF NEITHER IS SET
       76 A5     01     B1   00F9           CMPW     #1,PHD$W_EXTDYNWS(R5)            ;POSSIBLE TO LOCK 2 MORE PAGES?
                 08     1F   00FD           BLSSU    62$                              ;BRANCH IF YES
      50   0194  8F     3C   00FF           MOVZWL   #SS$_LKWSETFUL,R0                ;LOCKED PORTION OF WORKING SET FULL
              0084      31   0104           BRW      110$                             ;
       15 58     04     E1   0107  62$:     BBC      #WSL$V_PFNLOCK,R8,68$            ;BRANCH IF LOCKING IN WORKING SET
       0B 58     09     E1   010B           BBC      #LCK_V_GLOBAL,R8,64$             ;BRANCH IF PAGE IS NOT GLOBAL
                             010F                    PFN_REFERENCE   -
                             010F           ACBW     <#1,#1,@W^PFN$Ax_WSLX[R0],64$ >,- ;BRANCH IF FIRST LOCK IN
                             010F                    LONG_OPCODE=ACBL,-
                             010F                    IMAGE=SYS_NONPAGED
                             0118                                                     ;MEMORY OF THIS GLOBAL PAGE
                 06     11   0118           BRB      68$
           0000'CF      D7   011A  64$:     DECL     W^MMG$GL_PFNLOCK                 ;CAN ANY MORE PAGES BE LOCKED IN MEMORY?
                 71     19   011E           BLSS     150$                             ;BRANCH IF NO, TOO MANY ALREADY LOCKED
            63   58     88   0120  68$:     BISB     R8,(R3)                          ;SET NEW LOCK BIT IN WSLE
                 6E     95   0123           TSTB     (SP)                             ;WAS ENTRY ALREADY IN LOCKED PART OF WS?
                 5F     12   0125           BNEQ     100$                             ;BRANCH IF YES
                             0127  ;
                             0127  ; ENTRY MUST BE PLACED IN THE LOCKED PORTION OF THE WORKING SET
                             0127  ;
                 7C     10   0127           BSBB     SWAPLOCK                         ;SWAP THE WORKING SET LIST ENTRIES
                 62     96   0129           INCB     (R2)                             ;COUNT ANOTHER LOCKED WSLE FOR THIS PAGE TAB
                 03     14   012B           BGTR     70$                              ;BRANCH IF NOT THE FIRST IN THIS PT
              6C A5     B6   012D           INCW     PHD$W_PTCNTLCK(R5)               ;ANOTHER PAGE TABLE WITH LOCKED WSLE'S
              0E A5     B6   0130  70$:     INCW     PHD$W_WSDYN(R5)                  ;DYNAMIC PART OF LIST
                             0133                                                     ;STARTS ONE ENTRY HIGHER
  0E A5     10 A5       B1   0133           CMPW     PHD$W_WSNEXT(R5),PHD$W_WSDYN(R5) ;IS WSNEXT NO LONGER
                             0138                                                     ;POINTING INTO DYNAMIC REGION?
                 49     1E   0138           BGEQU    98$                              ;BRANCH IF IT'S OK
  10 A5     12 A5       B0   013A           MOVW     PHD$W_WSLAST(R5),PHD$W_WSNEXT(R5) ;OTHERWISE POINT IT TO END
                 42     11   013F           BRB      98$                              ;EXIT SUCCESSFULLY
                             0141  ;
```

```
                              0141   517 ; UNLOCK PAGE FROM WORKING SET LIST
                              0141   518 ;
              63   58    93   0141   519 80$:    BITB    R8,(R3)                    ;IS THIS PAGE LOCKED?
                   07    12   0144   520         BNEQ    90$                        ;BRANCH IF YES
    58    10   10  01    F0   0146   521         INSV    #SS$_WASCLR,#16,#16,R8     ;AT LEAST ONE PAGE WAS ALREADY UNLOCKED
                   39    11   014B   522         BRB     100$
           13 63   04    E1   014D   523 90$:    BBC     #WSL$V_PFNLOCK,(R3),94$    ;BRANCH IF PAGE NOT LOCKED IN MEMORY
           0F 58   04    E1   0151   524         BBC     #WSL$V_PFNLOCK,R8,94$      ;BRANCH IF NOT UNLOCKING IT FROM MEMORY
           07 58   09    E1   0155   525         BBC     #LCK_V_GLOBAL,R8,92$       ;BRANCH IF NOT A GLOBAL PAGE
                              0159   526                 PFN_REFERENCE   -
                              0159   527         DECW    <@W^PFN$Ax_WSLX[R0]>,-             ;ONE LESS MEMORY LOCK FOR THIS GLOBA
                              0159   528                 LONG_OPCODE=DECL,-
                              0159   529                 IMAGE=SYS_NONPAGED
                   04    14   015E   530         BGTR    94$                        ;BRANCH IF NOT LAST MEMORY LOCK
             0000'CF     D6   0160   531 92$:    INCL    W^MMG$GL_PFNLOCK           ;ANOTHER PAGE MAY BE LOCKED IN MEMORY
              63   58    8A   0164   532 94$:    BICB    R8,(R3)                    ;CLEAR THE DESIRED LOCK BIT(S)
              63   30    93   0167   533         BITB    #<WSL$M_WSLOCK ! WSL$M_PFNLOCK>,(R3) ;MUST PAGE REMAIN IN THE
                              016A   534                                            ;LOCKED PORTION OF THE WORKING SET?
                   1A    12   016A   535 96$:    BNEQ    100$                       ;BRANCH IF YES, IT IS STILL LOCKED
                              016C   536 ;
                              016C   537 ; MOVE THIS WORKING SET LIST ENTRY OUT OF THE LOCKED PORTION OF THE WORKING ET
                              016C   538 ;
                OE A5    B7   016C   539         DECW    PHD$W_WSDYN(R5)            ;INCLUDE LAST LOCKED SLOT IN DYNAMIC REGION
                   34    10   016F   540         BSBB    SWAPLOCK                   ;SWAP ENTRIES WITH THE LAST LOCKED ONE
                   52    DD   0171   541         PUSHL   R2                         ;SAVE BYTE INDEX OF PAGE TABLE PAGE
         52     08 AE    D0   0173   542         MOVL    8(SP),R2                   ;RESTORE VA
                 FE86'   30   0177   543         BSBW    MMG$SVAPTECHK              ;GET R3 = SVAPTE
                   52  8ED0   017A   544         POPL    R2                         ;RESTORE BYTE INDEX OF PAGE TABLE PAGE
              50   01    CE   017D   545         MNEGL   #1,R0                      ;INDICATOR DECREMENT OF PTWSLELCK ARRAY
                 FE7D'   30   0180   546         BSBW    MMG$MOVPTLOCK1             ;ONE LESS REASON PT PAGE MUST BE LOCKED
                 FE7A'   30   0183   547 98$:    BSBW    MMG$EXTRADYNWS             ;RECALCULATE EXTRA DYNAMIC WSLE COUNT
    50    58    F0 8F    78   0186   548 100$:   ASHL    #-16,R8,R0                 ;RETURN WASSET OR WASCLR
                   06    BA   0188   549 110$:   POPR    #^M<R1,R2>                 ;RESTORE R2 = VIRTUAL ADDRESS
                              018D   550         ENBINT                             ;AND RESTORE THE CALLING IPL
                         05   0190   551         RSB                                ;AND RETURN
                              0191   552
                              0191   553 ;
                              0191   554 ; TOO MANY PAGES LOCKED IN MEMORY TO ALLOW ANOTHER TO BE LOCKED
                              0191   555 ;
           05 58   09    E1   0191   556 150$:   BBC     #LCK_V_GLOBAL,R8,155$      ;BRANCH IF PAGE NOT GLOBAL
                              0195   557                 PFN_REFERENCE   -
                              0195   558         DECW    <@W^PFN$Ax_WSLX[R0]>,-             ;THIS GLOBAL PAGE NOT LOCKED IN MEMO
                              0195   559                 LONG_OPCODE=DECL,-
                              0195   560                 IMAGE=SYS_NONPAGED
             0000'CF     D6   019A   561 155$:   INCL    W^MMG$GL_PFNLOCK           ;FIX THE COUNTER
         50    00D4 8F   3C   019E   562         MOVZWL  #SS$_LCKPAGFUL,R0          ;NO MORE PAGES MAY BE LOCKED IN MEMORY
                   E6    11   01A5   563         BRB     110$
                              01A5   564
                              01A5   565         .DSABL  LSB
                         01A5              .SBTTL  SWAPLOCK - INTERFACE TO SWAPWSLE FROM LOCK/UNLOCK
                         01A5
                         01A5    ;
                         01A5    ; INPUTS:
                         01A5    ;
                         01A5    ;         R0 = PFN OF THE FIRST WORKING SET LIST SLOT
                         01A5    ;         R1 = INDEX TO FIRST WORKING SET LIST SLOT (WSLX1)
                         01A5    ;         R2 = BYTE INDEX OF PAGE TABLE CONTAINING THE PAGE BEING LOCKED/UNLOCKED
                         01A5    ;         R4 = PCB ADDRESS
                         01A5    ;         R5 = PROCESS HEADER ADDRESS - P1 OR SYSTEM SPACE
                         01A5    ;         IPL = SYNCH, ESSENTIAL FOR MANIPULATING WORKING SET LIST
                         01A5    ;                 FOR SYSTEM SPACE, AND BECAUSE PFN DATA BASE IS CHANGED
                         01A5    ;
                         01A5    ; OUTPUTS:
                         01A5    ;
                         01A5    ;         R0 PRESERVED
                         01A5    ;         R2 = ADDRESS OF BYTE CONTAINING COUNT OF LOCKED WORKING SET LIST
                         01A5    ;                 ENTRIES IN THE PAGE TABLE
                         01A5    ;         SPECIFIED WORKING SET LIST ENTRIES ARE SWAPPED
                         01A5    ;
                         01A5    SWAPLOCK:
          52       DD    01A5              PUSHL   R2                          ;SAVE PAGE TABLE NUMBER
     52   0E A5    3C    01A7              MOVZWL  PHD$W_WSDYN(R5),R2          ;INDEX TO SECOND WORKING SET LIST SLOT (WSLX
          09       10    01AB              BSBB    MMG$SWAPWSLE                ;SWAP THE WORKING SET LIST ENTRIES
52   55   64 A5    C1    01AD              ADDL3   PHD$L_PTWSLELCK(R5),R5,R2 ;BASE ADDRESS OF ARRAY OF COUNTS OF
                         01B2                                                  ;LOCKED WORKING SET LIST ENTRIES
        52   8E    C0    01B2              ADDL    (SP)+,R2                    ;ADDRESS OF COUNT BYTE FOR # OF LOCKED
                         01B5                                                  ;WSLE'S IN THE PAGE TABLE
                   05    01B5              RSB
```

```
0186           .SBTTL  SWAPWSLE - SWAP WORKING SET LIST ENTRIES
0186    ;++
0186    ; FUNCTIONAL DESCRIPTION:
0186    ;
0186    ;       THIS ROUTINE SWAPS THE CONTENTS OF THE FIRST WORKING SET LIST SLOT
0186    ; WHICH MUST BE ACTIVE WITH THAT OF THE SECOND WORKING SET LIST SLOT
0186    ; WHICH MAY BE EITHER ACTIVE OR FREE.
0186    ;
0186    ; CALLING SEQUENCE:
0186    ;
0186    ;       BSBW    MMG$SWAPWSLE
0186    ;
0186    ; INPUT PARAMETERS:
0186    ;
0186    ;       R0 = PFN OF THE FIRST WORKING SET LIST SLOT
0186    ;       R1 = INDEX TO FIRST WORKING SET  LIST SLOT (WSLX1)
0186    ;       R2 = INDEX TO SECOND WORKING SET LIST SLOT (WSLX2)
0186    ;       R4 = PCB ADDRESS
0186    ;       R5 = PROCESS HEADER ADDRESS - P1 OR SYSTEM SPACE
0186    ;       IPL = SYNCH, ESSENTIAL FOR MANIPULATING WORKING SET LIST
0186    ;               FOR SYSTEM SPACE, AND BECAUSE PFN DATA BASE IS CHANGED
0186    ;
0186    ; IMPLICIT INPUTS:
0186    ;
0186    ;       NONE
0186    ;
0186    ; OUTPUT PARAMETERS:
0186    ;
0186    ;       R0 PRESERVED
0186    ;
0186    ; IMPLICIT OUTPUTS:
0186    ;
0186    ;       NONE
0186    ;
0186    ; COMPLETION CODES:
0186    ;
0186    ;       NONE
0186    ;
0186    ; SIDE EFFECTS:
0186    ;
0186    ;       NONE
0186    ;
0186    ;--
```

```
                          01B6   ;
                          01B6   ; ****************************************************************
                          01B6   ;
                          01B6   ; *************** THE FOLLOWING CODE MUST BE RESIDENT ***************
                          01B6   ;
                      000001B6          .PSECT  $MMGCOD
                          01B6   ;
                          01B6   ; ****************************************************************
                          01B6   ;
                          01B6          .ENABL  LSB
                          01B6
                          01B6   MMG$SWAPWSLE::
         52   51   D1     01B6          CMPL    R1,R2                           ;IS THE SWAP A NOP?
              32   13     01B9          BEQL    90$                             ;BRANCH IF YES
            6541   DD     01BB          PUSHL   (R5)[R1]                        ;PUSH WSLE1
              05   BB     01BE          PUSHR   #^M<R0,R2>                      ;PUSH WSLX2, PFN(WSLX1)
         52 6542   D0     01C0          MOVL    (R5)[R2],R2                     ;GET WSLE2
              05   12     01C4          BNEQ    50$                             ;BRANCH IF NOT AN EMPTY ENTRY
            6541   D4     01C6          CLRL    (R5)[R1]                        ;JUST FREE WSLX1
              0F   11     01C9          BRB     70$                             ;AND GO SET UP WSLX2
           20 52   E9     01CB   50$:   BLBC    R2,SWAPWSLEBUG                  ;BRANCH IF WSLE NOT VALID, ERROR
              51   DD     01CE          PUSHL   R1                              ;SAVE WSLX1
           FE2D'   30     01D0          BSBW    MMG$SVAPTECHK                   ;RETURN R3 = SYS VA OF PAGE TABLE ENTRY
           FE2A'   30     01D3          BSBW    MMG$WSLEPFN                     ;GET PFN FROM SVAPTE OF THIS WSLE
              02   BA     01D6          POPR    #^M<R1>                         ;R1 = WSLX1
              02   10     01D8          BSBB    SETWSLE                         ;STORE WSLE AND WSLX
                          01DA   ;
                          01DA   ; MOVE THE OLD WSLE1 INTO SLOT INDEXED BY WSLX2
                          01DA   ; 0(SP) = PFN OF WSLE1, 4(SP) = WSLX2, 8(SP) = WSLE1
                          01DA   ;
              07   BA     01DA   70$:   POPR    #^M<R0,R1,R2>                   ;R0=PFN, R1=WSLX2, R2=WSLE1
                          01DC   ;
                          01DC   ; STORE WORKING SET LIST ENTRY INTO ITS WORKING SET LIST ENTRY SLOT
                          01DC   ; AND STORE WSLX INTO PFN DATA BASE IF PAGE TYPE IS NOT GLOBAL
                          01DC   ; R0 = PAGE FRAME NUMBER
                          01DC   ; R1 = WSLX, R2 = WORKING SET LIST ENTRY WITH PAGE TYPE
                          01DC   ;
                          01DC   SETWSLE:
       6541   52   D0     01DC          MOVL    R2,(R5)[R1]                     ;STORE WORKING SET LIST ENTRY
02  52   03   01   EC     01E0          CMPV    #WSL$V_PAGTYP,#WSL$S_PAGTYP,R2,#PFN$C_GLOBAL ;GLOBAL PAGE?
         06   18          01E5          BGEQ    90$                             ;BRANCH IF NOT
                          01E7                  PFN_REFERENCE   -
                          01E7          MOVW    <R1,@W^PFN$Ax_WSLX[R0]>,-       ;SET PFN WSLX
                          01E7                  LONG_OPCODE=MOVL,-
                          01E7                  IMAGE=SYS_NONPAGED
              05          01ED   90$:   RSB                                     ;AND RETURN
                          01EE   ;
                          01EE   ; LENGTH VIOLATION FOR VIRTUAL ADDRESS IN WORKING SET LIST
                          01EE   ;
                          01EE   SWAPWSLEBUG:
                          01EE          BUG_CHECK SWAPWSLE,FATAL
                          01F2
                          01F2          .DSABL  LSB
```

```
01F2              .SBTTL  SCNWSLX - SCAN FOR WORKING SET LIST INDEX
01F2      ;++
01F2      ;
01F2      ; FUNCTIONAL DESCRIPTION:
01F2      ;
01F2      ;       THIS ROUTINE SCANS THE ENTIRE WORKING SET FOR THE SPECIFIED
01F2      ; VIRTUAL ADDRESS.  IT IS CALLED AT IPL=SYNCH, LOWERS TO IPL=ASTDEL
01F2      ; TO DO THE SCAN AND RAISES BACK TO SYNCH WHEN COMPLETED.
01F2      ;       DROPPING IPL TO ASTDEL AND THUS MAKING THE PROCESS ELIGIBLE
01F2      ; FOR SWAPPING, PLACES THE FOLLOWING RESTRICTIONS ON THIS CODE AND
01F2      ; ITS CALLERS.
01F2      ;
01F2      ;       1) THIS CODE MAY NOT FAULT PROCESS PAGES SINCE THAT MIGHT
01F2      ; FAULT OUT THE DESIRED WORKING SET LIST ENTRY.
01F2      ;       2) THE CALLING CODE MAY NOT KEEP ANY SYSTEM SPACE ADDRESSES
01F2      ; FOR THE PROCESS HEADER OR PAGE TABLE ENTRIES SINCE THE HEADER BASE
01F2      ; ADDRESS MAY CHANGE IF THE PROCESS IS SWAPPED.
01F2      ;       3) THE CALLING CODE MAY NOT HOLD A PFN FOR A GIVEN PAGE SINCE
01F2      ; THAT TOO COULD CHANGE IF THE PROCESS WERE SWAPPED.
01F2      ;
01F2      ; CALLING SEQUENCE:
01F2      ;
01F2      ;       BSBW    MMG$SCNWSLX
01F2      ;
01F2      ; INPUTS:
01F2      ;
01F2      ;       R2  = VIRTUAL ADDRESS TO SCAN FOR
01F2      ;       R5  = P1 ADDRESS OF PROCESS HEADER
01F2      ;       IPL = SYNCH, DROPS TO ASTDEL, RESTORES TO SYNCH ON RETURN
01F2      ;             SEE FUNCTIONAL DESCRIPTION ABOVE!
01F2      ;
01F2      ; OUTPUTS:
01F2      ;
01F2      ;       R1  = WORKING SET LIST INDEX IF FOUND
01F2      ;           = 0 IF NOT FOUND
01F2      ;       R3  = PRESERVED
01F2      ;       IPL = IS LOWERED TO ASTDEL AND RESTORED TO SYNCH ON RETURN
01F2      ;             SEE FUNCTIONAL DESCRIPTION ABOVE!
01F2      ;--
```

```
                                 01F2  ;
                                 01F2  ; **********************************************************************
                                 01F2  ;
                                 01F2  ; **************** THE FOLLOWING CODE MUST BE RESIDENT *****************
                                 01F2  ;
                             000001F2          .PSECT  $MMGCOD
                                 01F2  ;
                                 01F2  ; **********************************************************************
                                 01F2  ;
                                 01F2  MMG$SCNWSLX::
                                 01F2          SETIPL  #IPL$_ASTDEL                    ;LOWER IPL FROM SYNCH TO ASTDEL
     52    52    F7 8F    78     01F5          ASHL    #-VA$S_BYTE,R2,R2               ;JUST INTERESTED IN PAGE NUMBER BITS
           51    0C A5    3C     01FA          MOVZWL  PHD$W_WSLOCK(R5),R1             ;INDEX TO FIRST NON-PERMANENT WSLE
              51   6541   DE     01FE          MOVAL   (R5)[R1],R1                     ;ADDRESS OF FIRST WSLE
             50    12 A5  3C     0202          MOVZWL  PHD$W_WSLAST(R5),R0             ;INDEX TO LAST (INCLUSIVE) WSLE
               50  6540   DE     0206          MOVAL   (R5)[R0],R0                     ;ADDRESS OF LAST WSLE
   52    61    17    09   EC     020A  20$:    CMPV    #VA$V_VPG,#VA$S_VPG,(R1),R2 ;IS THIS THE DESIRED WSLE?
                     0C   13     020F          BEQL    40$                             ;BRANCH IF YES
 FFF3 51    04       50   F1     0211          ACBL    R0,#4,R1,20$                    ;LOOP THROUGH ALL THE SLOTS
                                 0217  ;
                                 0217  ; WORKING SET LIST ENTRY NOT FOUND FOR SPECIFIED VIRTUAL ADDRESS
                                 0217  ;
                     51   D4     0217  30$:    CLRL    R1                              ;RETURN IMPOSSIBLE WSLX
                                 0219          SETIPL  #IPL$_SYNCH                     ;RESTORE IPL SYNCH
                          05     021C          RSB                                     ;AND RETURN TO CALLER
                                 021D  ;
                                 021D  ; FOUND THE WORKING SET LIST ENTRY
                                 021D  ;
                                 021D  40$:    SETIPL  #IPL$_SYNCH                     ;BACK TO IPL SYNCH
                F4 61     E9     0220          BLBC    (R1),30$                        ;MAKE SURE SWAPPER DIDN'T JUST
                                 0223                                                  ;RIP THE ENTRY OUT FROM UNDER US
              51     55   C2     0223          SUBL    R5,R1                           ;CONVERT ADDRESS
        51   51      1E   9C     0226          ROTL    #<32-2>,R1,R1                   ;BACK TO INDEX
                          05     022A          RSB                                     ;RETURN WITH WSLX IN R1
```

```
022B          .SBTTL  MMG$EXPKSTACK - EXPAND THE KERNEL STACK
022B    ;++
022B    ;
022B    ; FUNCTIONAL DESCRIPTION:
022B    ;
022B    ;       THIS ROUTINE EXPANDS THE KERNEL STACK BY ALLOCATING PHYSICAL MEMORY AND
022B    ;       USING THE RESERVED WORKING SET LIST ENTRIES.  THIS ROUTINE RETURNS WITH
022B    ;       SUCCESS IF THE STACK HAS ALREADY BEEN EXPANDED.
022B    ;
022B    ; CALLING SEQUENCE:
022B    ;
022B    ;       CALLS   #0,MMG$EXPKSTK
022B    ;
022B    ; INPUTS:
022B    ;
022B    ;       NONE
022B    ;
022B    ; OUTPUTS:
022B    ;
022B    ;       NONE
022B    ;--
```

```
                             022B   792 ;
                             022B   793 ; *****************************************************************
                             022B   794 ;
                             022B   795 ; **************** THE FOLLOWING CODE MUST BE RESIDENT *****************
                             022B   796 ;
                         0000022B   797         .PSECT  $MMGCOD
                             022B   798 ;
                             022B   799 ; *****************************************************************
                             022B   800 ;
                             022B   801 MMG$EXPKSTK::
                        007C 022B   802         .WORD    ^M<R2,R3,R4,R5,R6>
                             022D   803         DSBINT   #IPL$_ASTDEL              ;DON'T LET THINGS GET CONFUSED
           54   0000'CF   D0 0233   804         MOVL     W^SCH$GL_CURPCB,R4        ;R4 = PCB
     55   00000000'EF     D0 0238   805         MOVL     CTL$GL_PHD,R5             ;R5 = P1 PHD
          FFFFFFFF'8F     D0 023F   806         MOVL     #SWP$C_KSTACK_EX_WSL-     ;WSL INDEX FOR FIRST EXPANSION PAGE
                  56         0245   807                  +SWP$C_KSTACK_EX-1,R6
                6546      D5 0246   808         TSTL     (R5)[R6]                  ;HAVE WE ALREADY EXPANDED
                  7B      12 0249   809         BNEQ     30$                       ;YES
     52   00000000'8F     D0 024B   810         MOVL     #CTL$GL_KSTKBASEXP,R2     ;VA OF FIRST EXPANSION PAGE
                             0252   811 10$:    SETIPL   #IPL$_SYNCH
               FDA8'      30 0255   812         BSBW     MMG$SVAPTECHK             ;GET SVAPTE IN R3
           7E     52      7D 0258   813         MOVQ     R2,-(SP)
               FDA2'      30 025B   814         BSBW     MMG$FREWSLE               ;MAKE SURE WE CAN ADD A WSL PAGE
           52     6E      7D 025E   815         MOVQ     (SP),R2
              69  50      E9 0261   816         BLBC     R0,100$
       7E    10   A5      B0 0264   817         MOVW     PHD$W_WSNEXT(R5),-(SP)    ;SAVE THE FREE POINTER
        10  A5    56      D0 0268   818         MOVW     R6,PHD$W_WSNEXT(R5)       ;USE THIS WSL ENTRY
               FD91'      30 026C   819         BSBW     MMG$ININEWPFN             ;GET A PFN AND FILL IN WSL
        10  A5    8E      B0 026F   820         MOVW     (SP)+,PHD$W_WSNEXT(R5)    ;RESTORE THE FREE POINTER
           52     8E      7D 0273   821         MOVQ     (SP)+,R2
                  50      D5 0276   822         TSTL     R0
                  5E      19 0278   823         BLSS     110$                      ;NO PFN AVAILABLE
 00000000'FF40    07      88 027A   824         BISB     #PFN$C_ACTIVE,@L^PFN$AB_STATE[R0] ;MARK IT ACTIVE
         D0000000 8F      C9 0282   825         BISL3    #<PTE$M_VALID ! PTE$C_SRKW>,-
           63     50         0288   826                  R0,(R3)                   ;A VALID PTE
                             028A   827         INVALID R2
         6546     20      C8 028D   828         BISL     #WSL$M_WSLOCK,(R5)[R6]    ;SET LOCK BIT
              08  A5      B7 0291   829         DECW     PHD$W_QSLIST(R5)          ;WSL STARTS ONE PAGE SOONER
     52   0200    C2      9E 0294   830         MOVAB    512(R2),R2                ;NEXT PAGE
                             0299   831         SETIPL   #IPL$_ASTDEL              ;LET THE SYSTEM DO SOMETHING ELSE
 FFAE 56   FF 8F   00'8F  9D 029C   832         ACBB     #SWP$C_KSTACK_EX_WSL,#-1,R6,10$  ;NEXT WSL FOR EXPANSION
FFFFFFFC'EF  00000000'8F  D0 02A4   833         MOVL     #CTL$GL_KSTKBASEXP,CTL$AL_STACK-4 ; NEW STACK LIMITS
00000000'EF  00000000'8F  D0 02AF   834         MOVL     #CTL$GL_KSTKBASEXP,CTL$AL_STACKLIM
   0000'8F   00   6E  00  2C 02BA   835         MOVC5    #0,(SP),#0,#^X200*SWP$C_KSTACK_EX,-
             00000000'EF     02C1   836                  CTL$GL_KSTKBASEXP         ;ZERO THE NEW PAGES
                             02C6   837 30$:    ENBINT
           50     01      3C 02C9   838         MOVZWL #SS$_NORMAL,R0
                  04         02CC   839         RET
                             02CD   840 ;
               FD30'      30 02CD   841 100$:   BSBW     MMG$RESRCWAIT             ;PUT PROCESS ON CORRECT QUEUE
                  7E      DC 02D0   842 105$:   MOVPSL   -(SP)                     ;WAIT AT THIS PSL
               FD2B'      30 02D2   843         BSBW     MMG$SVPCTX                ;WAIT FOR RESOURCE
                FF7A      31 02D5   844         BRW      10$                       ;TRY AGAIN
                             02D8   845 ;
         50   0000'CF     7E 02D8   846 110$:   MOVAQ    W^SCH$GQ_FPGWQ,R0         ;FREE PAGE QUEUE
              08  A0      B6 02DD   847         INCW     WQH$W_WQCNT(R0)           ;ONE MORE WAITER
           60     64      0E 02E0   848         INSQUE   (R4),(R0)                 ;QUEUE THE PCB
```

```
2C A4   0A A0   B0  02E3          MOVW    WQH$W_WQSTATE(R0),PCB$W_STATE(R4) ;SET WAIT STATE IN PCB
           E6   11  02E8          BRB     105$
                    02EA
                    02EA          .END
```

```
...PFN              = 000001E7 R       04    PFN$C_PPGTBL        = 00000004
ACMODE              = 0000000C               PFN$C_PROCESS       = 00000000
BIT...              = 0000000A               PFN$C_SYSTEM        = 00000001
BUG$_SWAPWSLE         ********     X   04    PFN$S_PAGTYP        = 00000003
CTL$AL_STACK          ********     X   04    PFN$V_PAGTYP        = 00000000
CTL$AL_STACKLIM       ********     X   04    PHD$L_P0BR          = 000000C8
CTL$GL_KSTKBASEXP     ********     X   04    PHD$L_PSTBASOFF     = 00000020
CTL$GL_PHD            ********     X   03    PHD$L_PTWSLELCK     = 00000064
EXE$LCKPAG            0000001E RG      03    PHD$W_EXTDYNWS      = 00000076
EXE$LKWSET            00000017 RG      03    PHD$W_PTCNTLCK      = 0000006C
EXE$ULKPAG            00000007 RG      03    PHD$W_WSDYN         = 0000000E
EXE$ULWSET            00000000 RG      03    PHD$W_WSLAST        = 00000012
INADR               = 00000004               PHD$W_WSLIST        = 00000008
IPL$_ASTDEL         = 00000002               PHD$W_WSLOCK        = 0000000C
IPL$_SYNCH          = 00000008               PHD$W_WSNEXT        = 00000010
LCK_M_GLOBAL        = 00000200               PHD$W_WSQUOTA       = 00000018
LCK_M_LCKPAG        = 00000100               PHD$W_WSSIZE        = 00000050
LCK_V_GLOBAL        = 00000009               PR$_IPL             = 00000012
LCK_V_LCKPAG        = 00000008               PR$_TBIS            = 0000003A
MMG$CREDEL            ********     X   03    PRV$V_PSWAPM        = 0000000C
MMG$C_LENGTH        = FFFFFFE4               PTE$C_SRKW          = 50000000
MMG$EXPKSTK           0000022B RG      04    PTE$M_VALID         = 80000000
MMG$EXTRADYNWS        ********     X   04    PTE$S_OWN           = 00000002
MMG$FREWSLE           ********     X   04    PTE$S_PFN           = 00000015
MMG$GL_MAXPFN         ********     X   04    PTE$V_OWN           = 00000017
MMG$GL_PFNLOCK        00000000 RG      02    PTE$V_PFN           = 00000000
MMG$GL_SYSPHD         ********     X   04    PTE$V_TYP0          = 00000016
MMG$INADRINI          ********     X   03    PTE$V_WINDOW        = 00000015
MMG$ININEWPFN         ********     X   04    RETADR              = 00000008
MMG$LCKULKPAG         00000011 RG      04    SCH$GL_CURPCB         ********     X   04
MMG$L_MAXACMODE     = FFFFFFFC               SCH$GQ_FPGWQ          ********     X   04
MMG$MOVPTLOCK1        ********     X   04    SEC$V_RESIDENT        ********     X   04
MMG$RESRCWAIT         ********     X   04    SEC$W_FLAGS           ********     X   04
MMG$RETRANGE          ********     X   03    SETWSLE               000001DC R       04
MMG$SCNWSLX           000001F2 RG      04    SIZ...              = 00000001
MMG$SVAPTECHK         ********     X   04    SS$_ACCVIO          = 0000000C
MMG$SVPCTX            ********     X   04    SS$_LCKPAGFUL       = 000000D4
MMG$SWAPWSLE          000001B6 RG      04    SS$_LKWSETFUL       = 00000194
MMG$ULKGBLWSLE        00000000 RG      04    SS$_NOPRIV          = 00000024
MMG$WSLEPFN           ********     X   04    SS$_NORMAL          = 00000001
OP$_ACBL            = 000000F1               SS$_PAGOWNVIO       = 000001EC
OP$_ACBW            = 0000003D               SS$_WASCLR          = 00000001
OP$_DECL            = 000000D7               SS$_WASSET          = 00000009
OP$_DECW            = 000000B7               SWAPLOCK              000001A5 R       04
OP$_MOVL            = 000000D0               SWAPWSLEBUG           000001EE R       04
OP$_MOVW            = 000000B0               SWP$C_KSTACK_EX       ********     X   04
OP$_MOVZWL          = 0000003C               SWP$C_KSTACK_EX_WSL   ********     X   04
PCB$Q_PRIV          = 00000084               SYS$ADJWSL            ********     X   03
PCB$W_STATE         = 0000002C               ULKGBLWSLE            00000054 R       04
PFN$AB_STATE          ********     X   04    VA$S_BYTE           = 00000009
PFN$AB_TYPE           ********     X   04    VA$S_VPG            = 00000017
PFN$AL_BAK            ********     X   04    VA$V_VPG            = 00000009
PFN$AX_WSLX           ********     X   04    WQH$Q_WQCNT         = 00000008
PFN$C_ACTIVE        = 00000007               WQH$W_WQSTATE       = 0000000A
PFN$C_GBLWRT        = 00000003               WSL$M_PFNLOCK       = 00000010
PFN$C_GLOBAL        = 00000002               WSL$M_WSLOCK        = 00000020
PFN$C_GPGTBL        = 00000005               WSL$S_PAGTYP        = 00000003
```

```
WSL$V_PAGTYP                       = 00000001
WSL$V_PFNLOCK                      = 00000004
WSL$V_WSLOCK                       = 00000005

                                             +----------------+
                                             ! Psect synopsis !
                                             +----------------+

PSECT name                         Allocation          PSECT No.  Attributes
----------                         ----------          ---------  ----------
.  ABS  .                          00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                              00000000  (    0.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NCVEC BYTE
$$$210                             00000004  (    4.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC LONG
Y$EXEPAGED                         000000B1  (  177.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$MMGCOD                            000002EA  (  746.)  04 (  4.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
Z$INIT$PFN_FIXUP_TABLE             0000001E  (   30.)  05 (  5.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE

                                           +------------------------+
                                           ! Performance indicators !
                                           +------------------------+

Phase                      Page faults    CPU Time       Elapsed Time
-----                      -----------    --------       ------------
Initialization                      29    00:00:00.06    00:00:00.50
Command processing                 106    00:00:00.52    00:00:01.84
Pass 1                             384    00:00:13.59    00:00:32.50
Symbol table sort                    0    00:00:02.19    00:00:03.65
Pass 2                             162    00:00:03.15    00:00:06.25
Symbol table output                 16    00:00:00.12    00:00:00.22
Psect synopsis output                1    00:00:00.04    00:00:00.37
Cross-reference output               0    00:00:00.00    00:00:00.00
Assembler run totals               700    00:00:19.67    00:00:45.33

The working set limit was 1500 pages.
80740 bytes (158 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1392 non-local and 49 local symbols.
852 source lines were read in Pass 1, producing 34 object records in Pass 2.
31 pages of virtual memory were used to define 29 macros.

                                          +--------------------------+
                                          ! Macro library statistics !
                                          +--------------------------+

Macro library name                        Macros defined
------------------                        --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                  17
_$255$DUA28:[SYSLIB]STARLET.MLB;2                9
TOTALS (all libraries)                          26

1553 GETS were required to define 26 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SYSLKWSET/OBJ=OBJ$:SYSLKWSET MSRC$:SYSLKWSET/UPDATE=(ENH$:SYSLKWSET)+EXECML$/LIB
```

0386 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY